Stochastic Gradient Descent default hyperparameters:

'n0':1e-5,

'beta':1e-4,

'epsilon':1e-6,

'maxiters':10

NOTE: Average between training and validation must be weighted avg. (0.9\*training + 0.1\*valid)

Compare Closed Form and Stochastic Gradient Descent with default hyperparameters, with the 3 original features (No Text):

|  |  |  |
| --- | --- | --- |
| Metric | Closed Form | Gradient Descent |
| Time to run(s) | 0.239 | 0.335 |
| Validation MSE | 1.020 | 1.073 |
| Training MSE | 1.085 | 1.131 |
| Average(V&T) MSE |  |  |

Using SGD with default hyperparameters, without scaling text features, or taking out stopwords, compare:

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | No Text | Basic 160 Text Features | Basic 60 Text Features |
| Time to run | 0.335 | 16.090 | 5.303 |
| Validation MSE | 1.073 | 367.276 | 78,610.499 |
| Training MSE | 1.131 | 129,927.223 | 68,223.786 |
| Average (V&T) MSE |  |  |  |

Using SGD with default hyperparameters, with scaling text\_features so that they all fell within the 0 to 1 range, and multiplied that number by 6 to account for the overwhelming number of 0’s in the text\_features.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | No Text | Basic 160 Text Features | Basic 60 Text Features |
| Time to run(s) | 0.335 | 5.897 | 2.107 |
| Validation MSE | 1.073 | 1.0681 | 1.0679 |
| Training MSE | 1.131 | 1.1289 | 1.1291 |
| Average (V&T) MSE |  |  |  |

Using SGD with default hyperparameters, scaling factor above, and best above (160 text features), we now take out the stopwords.

|  |  |  |
| --- | --- | --- |
| Metric | No Stopwords | With Stopwords |
| Time to run(s) | 5.897 | 5.924 |
| Validation MSE | 1.0681 | 1.0713 |
| Training MSE | 1.1289 | 1.1261 |
| Average (V&T) MSE | 1.1228 | 1.1206 |

Improvement isn’t drastic, but it exists. From here on out we will not allow stopwords to be in the top 160 words we use.

Using all the preprocessing metrics above, we will now change some important hyperparameters to see how performance of above model in SGD changes.

Using best above, find the best configuration below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Alpha=  1e-8 | Alpha=1e-7 | Alpha\_0 = 1e-6 | Alpha\_0=1e-5 | Alpha\_0=1e-4 |
| Time to run(s) | 5.8314 | 5.9980 | 6.0039 | 5.9022 | 5.9990 |
| Validation MSE | 2.0661 | 2.0048 | 1.5668 | 1.0712 | 2.1e9 |
| Training MSE | 2.1056 | 2.0459 | 1.6178 | 1.1261 | 2.0e9 |
| Average(V&T) MSE |  |  |  |  |  |

For the purposes of this assignment, we don’t have time to exhaustively search the hyper parameters. It is possible that when alpha=1e-4, it actually converges for different values of Beta, but due to lack of time and tools, and it being vastly past the scope of the assignment, we will not further search for the complete pareto-optimal configuration.

**Hyper\_Parameter Optimization**

We will now take alpha=1e-5, our default value, and move forward. Now we try different values of Beta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Beta | 1e-2 | 1e-3 | 1e-4 | 1e-5 |
| Time to run(s) | 6.0937 | 6.3250 | 6.0229 | 5.7865 |
| Validation MSE | 1.0715 | 1.0712 | 1.0712 | 1.0712 |
| Training MSE | 1.1264 | 1.1261 | 1.1261 | 1.1261 |
| Average(V&T) MSE |  |  |  |  |

Taking Beta=1e-4, as it plateaus after:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterations | 10 | 100 | 1000 | 10000 |
| Time to run(s) | 5.7865 | 6.1525 | 7.4969 | 20.9270 |
| Validation MSE | 1.0712 | 1.0320 | 1.0438 | 1.0542 |
| Training MSE | 1.1261 | 1.0817 | 1.0579 | 1.0563 |
| Average(V&T) MSE |  |  |  |  |

Now that we have our optimal max\_iterations: 100 . After that, while the training data may be better, if we look at the INCREASING validation MSE, we can see that our model starts to overfit. From now on we will use 100 as our default hyperparameter for max\_iterations.

**New features**

All models considered here take the best parameter/preprocessing step taken from above unless otherwise specified.

Swear words:

We have text features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Original features | Ofeatures + swear word count | Ofeatures + Common word count | Ofeatures + Length of comments | Ofeatures + All new features |
| Time to run(s) | 6.1725 | 6.8806 | 7.3803 | 6.2892 | 8.0665 |
| Validation MSE | 1.0320 | 1.0282 | 1.0320 | 1.0324 | 1.0293 |
| Training MSE | 1.0817 | 1.0807 | 1.0817 | 1.0817 | 1.0806 |
| Average(V&T) MSE |  |  |  |  |  |

Yes, alone, it seems like only the swear word count decreases the mse of our regression… but check out what happens if we remove the text features completely from the original model, and keep the common word count instead.

|  |  |  |
| --- | --- | --- |
| Metric | Original features (no text) | Original features + Common word count |
| Time to run(s) | 0.1645 | 1.3892 |
| Validation MSE | 1.0301 | 1.0287 |
| Training MSE | 1.0935 | 1.0929 |
| Average(V&T) MSE |  |  |

As we can see, we get small boost in performance